Towards MPEG4: An Improved H.263-Based Video Coder *

Yuen-Wen Lee[1], Faouzi Kossentini [1], Rabab Ward[1], and Mark Smith [2]

[1]Department of Electrical Engineering,
University of British Columbia,
Vancouver BC V6T 1Z4, Canada

[2]School of Electrical and Computer Engineering,
Georgia Institute of Technology,
Atlanta, GA 30332-0250, USA

Abstract

This paper introduces a new H.263-based video coding algorithm for operation at very low bit rates. Although the algorithm is also based on block-based motion estimation/compensation and DCT coding, it is very different from conventional H.263-based algorithms. Our algorithm employs: 1) a rate-distortion-based mechanism to select amongst the H.263 16 × 16 macroblock coding types, 2) a fast median-based predictive motion searching technique, 3) a Lagrangian minimization for estimating the motion vectors, and 4) semi-fixed-length coders for coding the motion vectors and the DCT coefficients of the 8 × 8 motion-compensated prediction difference blocks. Experimental results indicate that the proposed algorithm significantly outperforms conventional H.263-based video coders both in terms of computational complexity and compression performance, while still producing a more error resilient bit stream. Another important advantage of our algorithm is that bit rate, quality, and number of computations can be controlled through manipulating the Lagrangian and threshold parameters. This feature is usually desired in many very low bit rate video communication applications due to power and mobility constraints.

*This work was supported by the Natural Sciences and Engineering Research Council of Canada under grant # OGP-0187668 and NASA.

1 Introduction

The great demand for very low bit rate video compression has motivated extensive research and standardization activities around the world. Many new compression algorithms have been developed that allow transmission or storage of QCIF resolution video with acceptable quality at bit rates as low as 16 kbps [1, 2, 3, 4]. Most notable are the H.263-based video coders [5], which have been shown recently to perform quite well in comparison with more complex video coders. The H.263 framework will likely be adopted by the MPEG4 group, which is expected to provide a toolkit-based audio-visual coding standard allowing many application-driven functionalities such as high compression performance and sufficient robustness in error-prone environments by the year 1998.

The emerging H.263 standard [5], like the MPEG ones, is based on motion-compensated prediction and DCT-based residual coding. However, it also involves more advanced prediction techniques, more effective motion vector coding methods, and more flexible mechanisms for alternating between inter and intra coding. These new features take into account the facts that, for very low bit rate video coding applications such as video telephony, video conferencing, and video monitoring, motion changes are relatively small and motion vector side information occupies a substantial portion of the overall bit rate.

While current H.263-based video compression algorithms appear to satisfy a number of existing applications, they do not support some of the eight key functionalities [6, 7, 8] sought by the MPEG4 group. In particular, both their compression performance and channel error robustness can be significantly improved. It is widely accepted that current H.263-based compression algorithms perform well for the target applications at bit rates between 20 and 64 kbps relative to coders of the earlier generation, but the compression performance tends to deteriorate rapidly at lower bit rates (such as 8 or 10 kbps). Moreover, although some safeguards are used, such as not coding the DC coefficient differentially, employing a median filter for motion vector prediction, and forcing intra coding every 132 16×16 macroblocks, there are still areas where significantly better channel error robustness can be achieved for only a small loss in compression performance.

The main reason the compression performance of most H.263-based video coders deteriorates at very low bit rates is that, as shown in Figure 1 for Telenor's H.263 coder [9], both the motion vector bit rate and the side information become excessive at such low rates. This is due to the following reasons:

- Inadequate macroblock (MB) coding control strategy: The mechanism used for switching between inter and intra coding is mostly ad-hoc.
- Independent estimation and coding of the motion vectors: Regardless of the target bit rate or quality, estimation and coding are performed independently.
- Inefficient motion estimation: The block-matching algorithm (BMA) usually used during the first step of motion estimation is generally inefficient. This is particularly acute when the sequence exhibits non-translational motion changes, and temporal changes due to occlusions, illumination variations, and zoom. Conventional BMAs tend to produce rough motion fields, containing high-entropy vectors that often do not contribute much to the motion-compensated prediction. In addition, such BMAs generally minimize the mean squared error (MSE) or the mean absolute difference (MAD), which do not necessarily result in the best video reconstruction quality. Finally, statistical redundancies and structure in the physical motion field are usually not well exploited.

In this papcr, we propose a H.263-based video coding algorithm that we developed to address the above issues. Our algorithm employs a rate-distortion (RD) based mechanism to select amongst the H.263's MB coding types. An RD criterion for alternating between intra and inter coding modes was first presented by Chung, Kossentini, and Smith [10] in a subband video framework. Similar criteria were then applied to Telenor's H.263 coder by Wiegand et al [11], by Chung et al [12], and by Schuster and Katsaggelos [13]. A common important result of the above investigations is the fact that significantly better RD tradeoffs can be obtained at very low bit rates using an RD mode selection criterion. The RD-based selection used in this work is also dependent, through the use of dynamic multipath searching, on the performance of the DCT residual coder. By exploiting this dependence, our algorithm becomes less sensitive to the type of measure (e.g., MSE, MAD, etc.) used to quantify distortion.

Our algorithm also employs a fast median-based predictive integer-pel motion estimation technique that minimizes the Lagrangian, i.e., the distortion [1] biased by the number of required bits. Recently, linear prediction has been proposed [14] to reduce the large computational complexity associated with BMA-based motion estimation. The application of several linear and nonlinear prediction techniques to BMA-based motion estimation is studied in [15]. We here study two types of median predictors: the H.263-specified 3-block median predictor and another 5-block median predictor. The Lagrangian minimization provides a more efficient cost measure that has also recently [16] been studied for motion compensation in the context of the H.261 framework. Our strategy, however, is based on our earlier work [10, 12], where the RD-based cost measure is analyzed in the context of a subband video coding framework.

Finally, the proposed algorithm employs semi-fixed-length coding techniques to encode both the motion vectors and the DCT coefficients. While we still use a median-based predictor similar to the one suggested by the H.263 standard, the prediction error is 2-layer fixed-length coded. Moreover, the DCT coefficients of the 8×8 motion-compensated prediction difference blocks are converted into events as described in the standard, but are then mapped into semi-fixed-length codes. These techniques can significantly increase the coder's robustness to channel errors. Their disadvantage, however, is that compatibility with current H.263 decoders is no longer achieved.

Besides its high compression performance and low computational complexity relative to Telenor's H.263 video coder, the proposed coding algorithm has some other important distinguishing features. First, our MB coding control strategy provides improved coding efficiency for arbitrary-shaped regions. Second, our motion analysis is based on the statistical behavior of the motion field; this reduces the estimation's sensitivity to non-motion temporal changes. Third, our median-predictive 2-layer fixed-length coding of the motion vectors and semi-fixed-length coding of the DCT coefficients can improve the robustness of the coder to channel errors. Finally, by manipulating the Lagrangian

[1] Although other distortion measures can be used, only the popular MSE and MAD will be considered.

parameter using feedback control techniques and by optimizing the threshold parameters, the rate, distortion, and computational complexity can be simultaneously controlled.

In the next section, we begin with a detailed description of the proposed approach, followed by a discussion of some practical issues. Experimental results and some conclusions are given in the last two sections.

2 Proposed Approach

As specified in the H.263 standard, intra coding of I-pictures consists of 8 × 8 DCT, uniform quantization, and then run-length and variable-length coding (or arithmetic coding). Left to the designer, however, is the problem of determining the value of the quantizer parameter QUANT (5 bits), which represents the quantization accuracy of the DCT coefficients. The only two requirements are that (1) the value of QUANT can be changed and transmitted only at the picture and/or the group-of-blocks (GOB) layers and (2) the four possible values of DQUANT (2 bits) are used to adjust the value of QUANT at the MB layer. We follow the H.263 approach, where QUANT of the first macroblock is set to the middle value (i.e., QUANT=16), and QUANT of each other macroblock is set during the encoding process to the value of QUANT of the previous macroblock. Then, given a Lagrangian parameter λ (whose value is obtained in Section 3) that controls the RD tradeoffs, the value of QUANT can be adjusted by one of the 4 possible values of DQUANT that minimizes $J_I^1(\lambda) = D_I + \lambda(R_I + 2)$, where R_I and D_I are the number of bits and the distortion (MSE or MAD) associated with the corresponding DCT coder, respectively. Next, we compare the smallest $J_I^1(\lambda)$ with $J_I^2(\lambda) = D_I + \lambda R_I$, the Lagrangian obtained in the case where DQUANT=0, and select the MB type that corresponds to the smaller value.

Inter coding of P-pictures depends on one previous picture (either a P- or an I-picture). For simplicity, PB-frames are not used in this work. For P-pictures, the basic coding operation consists of: 1) motion estimation and compensation, 2) MB type determination, 3) the possible coding of the motion vector(s), and 4) the possible DCT coding of the prediction difference blocks. This is the same operation defined in the H.263 standard.

For details about the H.263 coding steps, we refer the reader to [5]. We next present a general formulation of our coding control strategy, which is later adapted to the H.263 framework. This is followed by a description of a new median-based predictive motion vector estimation and coding method. We conclude this section by a discussion of our DCT coding approach in the context of the joint motion vector/DCT coding framework.

2.1 Macroblock coding control strategy

Assuming that the components of a video coding system are fixed and that the macroblocks within a P-picture are statistically independent, the optimal macroblock coding control strategy is the one that yields the best RD tradeoffs. More specifically, we should seek the motion vector $\mathbf{d} = (x, y)$ (if any) and the quality factor (QUANT) Q (if any) that minimize the Lagrangian

$$J(\lambda) = D_c(\mathbf{d}, Q) + \lambda \left[R_m + R_c + R_s\right]. \qquad (1)$$

In the above equation, $D_c(\mathbf{d}, Q)$ is the overall distortion, R_m is the number of bits needed to code the motion vector(s), R_c is the number of bits required for DCT coding, and R_s is the number of bits associated with side information. Table 1 provides a description of the parameters of equation (1) for all H.263's P-picture modes of operation. There are six modes of operation denoted by COD, INTER, INTER+Q, INTER4V, INTRA, and INTRA+Q. If the macroblock is not coded, the COD parameter is set to "1" and the current macroblock is replaced by the macroblock at the same spatial location in the previous reconstructed picture. In this case, only the COD parameter needs to be coded. This mode is designed for areas in the picture where little or no change relative to the previously reconstructed picture is detected. In both the INTER and the INTER+Q modes of operation, one motion vector is transmitted along with the DCT coefficients of the prediction difference blocks. The term D_c is the average DCT quantization error of the difference blocks. The difference between the INTER and INTER+Q modes of operation is that, in the latter, the value of QUANT is being changed. This is often required to compensate for prediction inaccuracies. The mode INTER4V is similar to the mode INTER, except that four motion vectors representing four 8 × 8 blocks are transmitted. The mode INTER4V is found to be useful for picture areas with high motion activity. Due to noise, occlusion, zoom, large illumination changes, or complex motion activity, simple translational motion-compensated prediction may be inadequate. In such a case, operating in the INTRA or INTRA+Q modes may be beneficial. The term D_c in both modes represent the DCT quantization error of the current macroblock. In the INTRA+Q mode, the parameter QUANT is being changed, usually to compensate for illumination variations.

It is clear from Table 1 that *six* Lagrangian values must be computed in order to find deterministically the coding mode that yields the lowest Lagrangian value. However, this is generally impractical especially that the INTER modes of operation involve the joint optimization between the motion vector estimation/coding and the DCT coding of the corresponding prediction difference blocks. That is, for every motion vector candidate, the DCT coder must be applied to the difference block(s) in order to evaluate both R_c and D_c. Moreover, as can be seen from equation (1), even the evaluation of D_c must be performed for all values of DQUANT. The latter part can be performed efficiently as the DCT is a unitary transformation (i.e., only *one* forward DCT needs to be evaluated for each 8×8 block). However, although the motion vector search area can be reduced substantially as described in the next subsection, the evaluation of the INTER Lagrangians can still be computationally demanding.

One way to reduce the total number of computations is to employ thresholding techniques that allow us to safely eliminate the INTRA and/or INTER coding options from consideration. For example, the Lagrangian of the COD mode of operation can be compared to a varying threshold T_n, and if it is smaller than T_n, the other modes are no longer considered. These techniques are similar, in principle, to those employed in Telenor's H.263 coder, but are here found to be more effective because the rate and distortion are more carefully traded. Although practical constraints often dictate that such methods be used, they are unfortunately still ad-hoc as there is no guarantee that the best coding control decision has been made.

An alternative way to reduce the computational complexity is to decouple the motion vector coding and DCT coding processes so that the Lagrangian minimization is applied sequentially. More specifically, we first locate the motion vector(s) that yield(s) the minimum Lagrangian, which is formed by the estimation distortion biased by the number of bits required for the coding procedure. This is described in detail in the following subsection. The corresponding difference blocks are then computed and DCT coded as discussed in Subsection 2.3. Finally, the resulting average values of rates and distortions, along with values of R_s and R_m, are used to compute the INTER mode Lagrangians. Of course, this procedure is sub-optimal. Thus, a potentially better solution is to maintain the sequential structure but apply a dynamic multipath searching (M-search) technique [17], where generally more than one motion vector can be considered as a good candidate, and DCT coding is applied to each of the corresponding difference macroblocks.

Dynamic M-search is similar to conventional M-search in the sense that M paths are considered as good candidates for the next stage. In dynamic M-search, however, the value of M is not fixed. It is varied from one macroblock/block to another, depending on the distribution and values of the Lagrangians associated with the candidate motion vectors. More specifically, only those motion vectors whose corresponding Lagrangian values are located within a small neighborhood centered at the smallest Lagrangian value are retained as good candidates. The neighborhood is contracted or expanded based on a computational constraint, as discussed in [17]. Dynamic M-search usually outperforms conventional M-search and also achieves a level of performance very close to that of full-search. However, as shown experimentally in the next section, it here yields only a slight performance advantage. As dynamic M-search is also very efficient, even such a small gain may be worth the additional complexity.

2.2 Motion Vector Estimation and Coding

For each macroblock and its four 8×8 luminance blocks in the current picture, a motion vector $\mathbf{d} = (x, y) \in \mathcal{S}$, where $\mathcal{S}$ is the set of all possible vectors in the search area, is sought. Each motion vector is chosen to minimize the Lagrangian

$$J_\lambda^m(\mathbf{d}) = \sum_{\mathbf{r} \in \mathcal{W}} \rho\left(I(\mathbf{r}, n) - I(\mathbf{r} + \mathbf{d}, n-1)\right) + \lambda \ R^m(\mathbf{d}), \tag{2}$$

where $\mathbf{r}$ is the spatial index of an image pixel, n is the time index, $I(\mathbf{r}, n)$ is the image intensity of the candidate macroblock/block in the current picture, $I(\mathbf{r} + \mathbf{d}, n-1)$ is the image intensity of the matching macroblock/block in the previous reconstructed picture, $\mathcal{W}$ is the size of the matching window, $\rho(\cdot)$ is the square operation (MSE) or the absolute operation (MAD), and $R^m(\mathbf{d})$ is the number of bits required to encode the motion vector. Minimizing $J^m_\lambda(\mathbf{d})$ is equivalent to the process of full-search entropy-constrained vector quantization, where the codebook contains all 16×16 (or 8×8) vectors in the search area. After determining the best motion vector(s) $\mathbf{d}^*$ representing the macroblock or each of the four 8×8 blocks, either one or four motion vectors, whichever yields better RD tradeoffs, is selected.

Many experimental results have shown that using $\frac{1}{2}$-pel accuracy motion estimation in our RD framework yields an insignificant improvement or a decrease in compression performance relative to integer-pel accuracy motion estimation at very low bit rates. Thus, only integer-pel accuracy is used; this slightly reduces the computational complexity. The latter can be reduced dramatically if the size of the set $\mathcal{S}$ and/or the size of the matching window $\mathcal{W}$ is reduced. Several techniques have been proposed [2, 18] that achieve this goal, at the expense of some loss in estimation performance. We introduce a median-based predictive searching technique that is similar in principle to the statistical-based predictive techniques described in [10, 19, 12, 20], but is simpler in concept and implementation. Our technique exploits the fact that, in very low bit rate applications, physical motion is usually very limited, structured, and slowly-varying. As illustrated in Figure 3, we first determine the most likely motion vector given a prediction model. Then, only the candidate motion vectors in the diamond-shaped small search area centered at the most likely motion vector is considered. Depending on the prediction model, the bit rate of operation and the content of the video scene, we found that full-search yields only a few $(2-5\ \%)$ motion vectors that do not belong to the diamond-shaped search area. Thus, such a technique can reduce substantially the number of computations at the expense of only a small loss in estimation performance.

If compatibility with current H.263-based decoders is desired, then the independent median-based prediction and motion vector difference variable-length coding methods described in the H.263 standard must be followed. However, we here propose an alternative method that is more robust to channel noise (Figures 4 and 5). Figure 4 compares the ROS's used by the H.263-specified prediction model and ours. Not only the 5-block ROS which includes 4 spatially and 1 temporally neighboring macroblocks/blocks, provides more prediction accuracy, but it also leads to a motion vector coder output that is more resilient to channel errors. In fact, it can be easily verified that, using the 5-block ROS, only three or more erroneously decoded motion vectors (as compared to two or more of them using the 3-block ROS) can cause error propagation.

Figure 5 illustrates our motion vector coding strategy. The motion vector consisting of the two median-predicted x and y components is placed at the center of the diamond-shaped search area. The probabilities shown in the figure are estimated using a very large training sequence. They represent the likelihood of choosing each of the motion vectors when a full-search BMA is employed. By ignoring motion vectors outside the search area, we can code efficiently each of the candidate motion vectors using either exactly *two* or exactly *four* bits as follows: the two most significant bits indicate one of the four layers shown in the figure. The code **00** indicates that the first layer (or the center of the search area) is selected, and no more bits are necessary. Otherwise, two more bits are necessary to indicate which of the four vectors in the selected layer is chosen. This semi-fixed-length coding technique is very simple, yet experiments show that its compression performance is as good as that of the H.263 variable-length technique. Moreover, assuming limited motion in the video sequence, constraining the 2-D range of the motion vectors will reduce sensitivity to video input noise. But the most important advantage of the proposed technique is its high robustness to channel errors. First, by providing more protection to the first two bits, the loss of synchronization,. which would have been likely using a H.263-compatible technique, can be avoided. Second, the effect of the channel errors can be reduced by applying gray coding as shown in the figure.

2.3 DCT Coding

According to the H.263 standard, the coding of the DCT coefficients in the INTRA mode of operation is performed in two steps. First, the DC coefficient is mapped to one of the levels of an 8-bit uniform quantizer whose output is fixed-length encoded. Second, the AC coefficients are converted into *events*, which are then mapped to variable-length codes (VLCs). The DC coefficient is coded non-differentially mainly to reduce the sensitivity of its corresponding bits to channel errors. In the INTER mode of operation, all DCT coefficients are translated into events, which are also mapped to VLCs.

An event is a combination of three parameters: LAST which indicates whether this is the last nonzero coefficient, RUN which is the number of successive zeros preceding the coded coefficient, and LEVEL which is the nonzero value of the coded coefficient. Table 12 of the H.263 standard provides the VLCs for the 101 most likely events. The remaining events are coded with a 22-bit word consisting of 7 bits for ESCAPE, 1 bit for LAST, 6 bits for RUN, and 8 bits for LEVEL.

The major problem associated with the above procedure is that a single channel bit error can propagate to many other 8 × 8 blocks, mainly due to the VLC procedure. Fortunately, our statistical analysis revealed that variable-length coding is neither required nor useful. In fact, we have constructed a 6-bit fixed-length code to which we map one of the 63 most likely events. The 64th code is set to ESCAPE so that the remaining events can be coded using the 22-bit word described earlier. Using the test sequences **MISS AMERICA** and **CAR PHONE**, only approximately 1% loss in compression performance was observed. The benefit of our coding procedure, however, is that loss of synchronization can be avoided and error propagation can be contained within a macroblock. We first apply pseudo-gray coding to the 6-bit codes so that those codes which are close in the Hamming distance sense represent events having the same values of LAST and RUN. This is because a channel bit error in LAST or RUN will cause error propagation while a similar error in LEVEL can be easily concealed. Understanding that pseudo-gray coding can only reduce the probability of error propagation, we also introduce a 5-bit macroblock header, where the first bit indicates the number of occurrences of ESCAPE and the four bits provide the number of occurrences of normal events. Of course, this implies that at most one ESCAPE and 16 normal events are allowed within a macroblock. According to our simulation results, this constraint is violated in less than 0.25 % of the time. In such an unlikely case, the decoder can either set the missing coefficients to zero or employ some estimation techniques. But the 5-bit header will eliminate error propagation, regardless of how many errors occur in the fixed-length codes.

The central goal of the proposed DCT coding procedure is to obtain a substantial improvement in channel error robustness at the expense of an insignificant loss in compression performance. The price paid is that our encoder's output bit stream is no longer compatible with conventional H.263-based video decoders. Thus, if compatibility is highly desired, the H.263-specified DCT coding procedure must still be used.

3 Practical Issues

As a consequence of the three-dimensional predictive coding technique and the MB coding control strategy, symbols representing the coded macroblocks/blocks for the current coder may not be available at the decoder. Thus, both the encoder and decoder must estimate these symbols using the same method. Our experimental results have shown that the simple H.263 procedure [5] consisting of four sequentially ordered decision rules is adequate for practical purposes. Thus, we decided to adopt such a procedure, especially that it improves our coder's compatibility with current H.263 decoders.

The most important practical issue is finding appropriate values for the Lagrangian parameter λ and the threshold T_n, the two key parameters that control the performance-computation tradeoffs. Although they are generally not independent, we here optimize them separately. The Lagrangian parameter λ controls mainly the RD tradeoffs. One method to find a particular value for λ is to employ the bisection search algorithm described in [21]. This method, however, is usually computationally demanding, as a large number of iterations may be required. An alternative method is to update λ using least-mean-squares (LMS) adaptation, as adopted in [11]. In this work, the parameter λ is initially estimated based on computed long-term statistics, and is then varied adaptively during the encoding process depending on a rate and/or a quality constraint, following the simple linear feedback control algorithm discussed in [22]. Without loss of generality, suppose we are operating in a fixed-rate communication environment [2], where a very

[2] Note that, in some variable-rate applications, constant reproduction quality is often desired.

slowly varying or constant video encoder throughput is desired. For this purpose, a sufficiently large memory buffer of size S_{max} is allocated. Then, the time-dependent size $s(t)$ of the buffer can be determined by the recursion $s(t+1) = s(t) + R(t) - B$, where $R(t)$ is the variable output bit rate of the encoder and B is the fixed output rate of the buffered contents. It is desirable that $s(t)$ be as close as possible to $s^* = \frac{S_{max}}{2}$. Based on the linear feedback control strategy of [22], it is assumed that, at time t, the parameter $\lambda(t)$ and the buffer size $s(t)$ are related by $\lambda(t) = c\ s(t)$, where the value of c depends on the RD characteristics of the video signal. For simplicity, it is also assumed that the video source is stationary, and that the operating distortion-rate curve is represented by $D(R) = A\alpha^{-R}$ for $R > 0$, where the parameters $A > 0$ and $\alpha > 1$ are determined based on the statistics of the video sequence. Then, the parameter c can be expressed as a function of A, α, and B as follows,

$$c = \frac{2A \log_e \alpha \ \ \alpha^{-B}}{S_{max}}.$$

Since the video sequence is generally not stationary, a new value for c is here determined for each frame based on the local RD characteristics. Using the above model, λ is increased or decreased linearly, where the rate of change c takes into account the nonlinear RD relationship. Relative to more general algorithms described in [22], the above algorithm is very simple. Its performance, which is illustrated in the next section, is relatively good. In fact, using a buffer of size 10 kilobits, the problem of overflow/underflow was never encountered during our coding simulations.

The threshold T_n is used to provide a good balance between performance and number of computations. A small value of T_n increase both the computational complexity and the probability of achieving the optimal solution. A large value of T_n can reduce dramatically the computational complexity but at the expense of possibly selecting an inferior operating mode. The H.263 standard suggests some constant thresholds that have been optimized to incorporate the MAD as a cost measure and to favor the predicted motion vector. Although rate and distortion are taken into account simultaneously in our framework, similar thresholds can still be heuristically derived. To improve our estimation of the threshold T_n, we also incorporate memory in the form of simple prediction models indicating the level of activity in the video scene.

4 Experimental Results

The following experimental results illustrate the computational complexity and performance of the proposed H.263-based video coding algorithm at very low bit rates. As in Telenor's coder, motion estimation and compensation is performed only for the Y luminance component. The estimated motion vector field is subsequently used for the motion compensation of the Cr and Cb chrominance signals. However, only integer-pel accuracy motion estimation is allowed in our framework[3].

We compared the coder's performance and complexity with that of Telenor's H.263 video coding implementation [9] using the QCIF test sequences **MISS AMERICA** and **CAR PHONE**. For fairness, the options/parameters of both coders are set to the same values. For example, neither implementation employs PB-frames. Moreover, the frame rate is set to 10 frames per second, advanced prediction is used, and the unrestricted motion vector mode is selected. Finally, the MSE is used as our coder's distortion measure in most experiments. The only exception is the first set of experiments, where both the MAD and the MSE are employed.

Figure 6 compares the average peak-signal-to-noise-ratio (PSNR) results of our coder with those of Telenor's H.263 coder for 150 frames of the color test video sequence **MISS AMERICA** at bit rates between 4 and 10 kilobits per second (kbps). The only difference between the two coders is the MB coding control strategy. We select the best mode of operation based on an RD criterion, as described previously. As seen from Figure 6, our strategy yields a significant improvement in PSNR, especially at very low bit rates. This is expected because while the motion bit rate in Telenor's coder is nearly constant (see Figure 1), it is reduced at lower bit rates (see Figure 7) by using the RD-based mode selection criterion. Such a criterion yields a more efficient allocation of bits amongst the coder's components. It is, however, more demanding in terms of computations. Nevertheless, by applying the thresholding techniques discussed earlier,

[3] Note that compatibility with H.263 decoders can still be maintained.

we have obtained an insignificant loss in PSNR while requiring only 5 % of the number of computations needed to determine all six Lagrangian values.

Figure 6 also shows that, when the MAD is substituted for the MSE, the PSNR gain is reduced by $10-25$ %. Since the PSNR is inversely proportional to the MSE, minimizing the MSE is equivalent to maximizing the PSNR. Thus, using a non-MSE measure such as the MAD generally yields a slightly lower PSNR. The advantage of using the MAD, however, is that the computational complexity is significantly reduced.

Figure 8 illustrates that our fast integer-pel accuracy motion vector estimation and coding compares very favorably in terms of PSNR with the full-search motion estimation and variable-length coding used in the earlier implementation. Moreover, as demonstrated earlier and confirmed by our simulations, our technique reduces the number of computations required for motion estimation by more than one order of magnitude. This is an important feature because motion estimation usually requires the lion's share of the computational load.

Also illustrated in Figure 8 is the fact that the new semi-fixed-length motion vector coding technique yields an insignificant loss in PSNR. Moreover, besides its lower complexity, semi-fixed-length coding can significantly improve the bit stream's resilience to channel errors. This is illustrated in Figure 9, which shows that our technique can lead to a much more graceful quality degradation as a function of increasing bit error rate. Recall that we divide the motion vector code into two parts: a 2-bit header and a 2-bit extension code. If both the header and the extension codes are not protected, the motion vector bits (like the VLC bits) are very sensitive to channel errors. If only the header is protected by, for example, a convolutional channel coder, the PSNR decreases very slowly. One can argue that VLC codes can be similarly protected. However, approximately twice the number of additional channel coder bits would be required since the number of header bits is slightly more than 50 % of the number of VLC bits. Since error propagation is found experimentally to be the major source of quality degradation in the presence of channel errors, the objective should be to minimize the number of required header bits. Although our technique employs only 2 bits per motion vector as header, other techniques using a 1-bit header may be more beneficial. This is a subject for further research.

As a practical solution to the joint motion vector estimation/coding and DCT residual coding, our proposed dynamic M-search technique improves the PSNR performance, as is shown in Figure 10. The improvement is not substantial, but the additional complexity is also relatively small. The slight PSNR gain suggests that, at very low bit rates, the MSE used as part of the Lagrangian cost measure is a good approximation of the 8×8 DCT coder's MSE.

Next, we present simulation results that illustrate the effectiveness of the feedback technique used to control the bit rate or quality through varying the parameter λ. Figures 11 and 12 show the bit rate and PSNR profiles of our coder when constant reproduction quality and bit rate (respectively) are placed as constraints on the encoding algorithm. Although a simple model is used, the feedback control technique is very effective. Notice that the PSNR in Figure 11 is nearly constant for all frames, indicating that it is easier to control the output quality level. This can be beneficial in several applications where a constant quality of service is highly desired. However, Figure 12 indicates that it is more difficult to control the bit rate. To reduce the likelihood of overflow/underflow, a sufficiently large buffer should be used. However, if the buffer size is constrained, employing more sophisticated feedback control techniques or incorporating memory into the bit rate control algorithm [22] may provide better solutions.

Figures 13(a) and 13(b) illustrate the overall PSNR improvement of our coder over Telenor's H.263 for the two test sequences **MISS AMERICA** and **CAR PHONE**, respectively. Our coder differs from Telenor's in that it incorporates RD-based control, thresholding, median-based predictive motion estimation, semi-fixed-length motion vector and DCT coding, and dynamic M-search. Not only does our coder outperform Telenor's, especially at the lower bit rates, but our coder's computational complexity is also lower even using the MSE. Notice that, although semi-fixed-length coding is employed, we still obtain a significant improvement in PSNR performance. But more important is the fact that our coder's subjective quality is superior to that of Telenor's coder. For example, when many viewers were presented with 150 frames of the decoded color sequence **MISS AMERICA**, which was encoded using the new coder at 4 kbps, they all reported that the subjective quality is either acceptable or good. When presented with another 150 frames produced by decoding Telenor's H.263 output bit stream, the viewers stated that the quality is both unacceptable and inferior to our coder's subjective quality.

Finally, to illustrate the behavior of our coder at higher bit rates, the x-axis of Figure 13(a) is extended to 16 kbps and that of Figure 13(b) is extended to 40 kbps. As the proposed techniques are designed for the very low bit rate range of operation, it should not be surprising that the PSNR gap decreases with increasing bit rate. For example, the RD-based MB mode selection criterion makes better use of the available bits, but the improvement is significant only when the bit budget is limited. Another example is $\frac{1}{2}$-pel motion estimation accuracy, which becomes beneficial at higher bit rates, but is not used in this framework. However, most of the proposed techniques can be optimized so that the performance improvement is large over a wide range of bit rates. This is subject for further research.

5 Conclusions

We have presented a new H.263-based video coder that provides an RD-based mechanism for alternating amongst H.263's modes of operation, and employs median-based predictive motion estimation/coding. We have also introduced two new semi-fixed-length coding techniques: one for coding the motion vectors and another for coding the DCT coefficients of the prediction difference blocks. We have demonstrated that only a small loss in compression performance is sacrificed for a likely significant increase in channel error robustness and a reduction of complexity. The only disadvantage is that the above techniques produce a bit stream that is not necessarily decodable by current H.263 video decoders.

Our coder outperforms Telenor's H.263-based video coder in compression performance and complexity, simultaneously. It also offers the user more control over the bit rate, quality, and number of computations, and it can produce a bit stream that is more resilient to channel errors.

Although we have not addressed many other issues such as content-based scalability and temporal random access, we believe that the proposed techniques provide some efficient solutions to two desired MPEG4 functionalities: improved coding efficiency and high robustness in error-prone environments.

6 Acknowledgment

The authors would like to thank Mr. Hong Man for his help during the fixed-length coding simulation work. We would also like to thank the anonymous reviewers for their helpful and constructive comments and suggestions. In particular, special thanks are extended to the reviewer who has provided us very helpful information on the status of the MPEG4 standardization effort.

References

[1] K.-H. Tzou, H. G. Musmann, and K. Aizawa, "Special Issue on Very Low Bit Rate Video Coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, pp. 213–367, June 1994.

[2] B. Girod, D. J. LeGall, M. I. Sezan, M. Vetterli, and Y. Hiroshi, "Special issue on image sequence compression," *IEEE Trans. on Image Processing*, vol. 3, Sept. 1994.

[3] W. Li, Y.-Q. Zhang, and M. L. Liou, "Special Issue on Advances in Image and Video Compression," *Proc. of the IEEE*, vol. 83, pp. 135–340, Feb. 1995.

[4] D. Anastassiou, "Current status of the MPEG-4 standardizaton effort," in *SPIE Proc. Visual Communications and Image Processing*, vol. 2308, pp. 16–24, 1994.

[5] ITU Telecom. Standardization Sector Study Group 15, "Document LBC-95 Working Party 15/1 Question 2/15," *Draft Recommendation H.263*, Leidschendam, 7 April 1995.

[6] ISO-IEC, "MPEG4 proposal package description - revision 3," *JTC1/SC2/WG11*, 1995.

[7] F. Pereira, "MPEG4: a new challenge for the representation of the audio-visual information," in *Proceedings of the International Picture Coding Symposium*, March 1996.

[8] C. Reader, "MPEG4 syntactic descriptive language: a universal interface for exchange of coded audio-visual data," in *Proceedings of the International Picture Coding Symposium*, March 1996.

[9] Telenor Research, "TMN (H.263) encoder / decoder, version 1.4a," *TMN (H.263) codec*, May 1995.

[10] W. Chung, F. Kossentini, and M. Smith, "A new approach to scalable video coding," in *IEEE Data Compression Conference*, (Snowbird, UT, USA), pp. 381–390, Mar. 1995.

[11] T. Wiegand, M. Lightstone, D. Mukherjee, T. Campbell, and S. Mitra, "Rate-Distortion Optimized Mode Selection for Very Low Bit Rate Video Coding and the Emerging H.263 Standard," *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 182–190, Apr. 1996.

[12] W. Chung, F. Kossentini, and M. Smith, "An efficient motion estimation technique based on a rate-distortion criterion," in *ICASSP96*, (Atlanta, GA), May 1996.

[13] A. Schuster and A. Katsaggelos, "Fast and efficient mode and quantizer selection in the rate-distortion sense for h.263," in *SPIE Proc. Visual Communications and Image Processing*, 1996.

[14] R. Arminato, R. Schafer, F. Kitson, and V. Bhaskaran, "Linear predictive coding of motion vectors," in *Proceedings of the IS&T/SPIE EI'96*, Jan. 1996.

[15] Y. Lee, F. Kossentini, M. Smith, and R. Ward, "Predictive rd-constrained motion estimation for very low bit rate video coding," *Submitted to the Special Issue of the IEEE Transactions on Selected Areas in Communications*, Aug. 1996.

[16] D. Hoang, P. Long, and J. Vitter, "Efficient cost measures for motion compensation at low bit rates," in *IEEE Data Compression Conference*, (Snowbird, UT, USA), pp. 102–111, Apr. 1996.

[17] F. Kossentini and M. Smith, "A fast searching technique for residual vector quantizers," *Signal Processing Letters*, vol. 1, pp. 114–116, July 1994.

[18] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architecture.* Boston: Kluwer Academic Publishers, 1995.

[19] W. Chung, F. Kossentini, and M. Smith, "Rate-distortion constrained statistical motion estimation for video coding," in *ICIP95*, (Washington, DC), Oct. 1995.

[20] F. Kossentini, W. Chung, and M. Smith, "Rate-distortion-constrained subband video coding," *Submitted to Transactions on Image Processing*, Mar. 1996.

[21] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. on Image Processing*, vol. 2, pp. 160–174, April 1993.

[22] J. Choi and D. Park, "A stable feedback control of the buffer state using the Lagrangian multiplier method," *Special Issue on Image Sequence Compression*, vol. 3, pp. 546–558, Sept. 1994.

List of Tables

List of Figures

MODE	COD	MB Type	Q	R_m	R_c	R_s
Not coded	1	N/A	N/A	N/A	N/A	COD
INTER	0	0	QUANT	MVD	DCT/RES	COD+INTER
INTER+Q	0	1	DQUANT	MVD	DCT/RES	COD+INTER+Q
INTER4V	0	2	QUANT	MVD2-4	DCT/RES	COD+INTER4V
INTRA	0	3	QUANT	N/A	DCT/INTRA	COD+INTRA
INTRA+Q	0	4	DQUANT	N/A	DCT/INTRA	COD+INTRA

Table 1: Values of COD, MB Type, Q, R_m, R_c, and R_s for various coding modes.

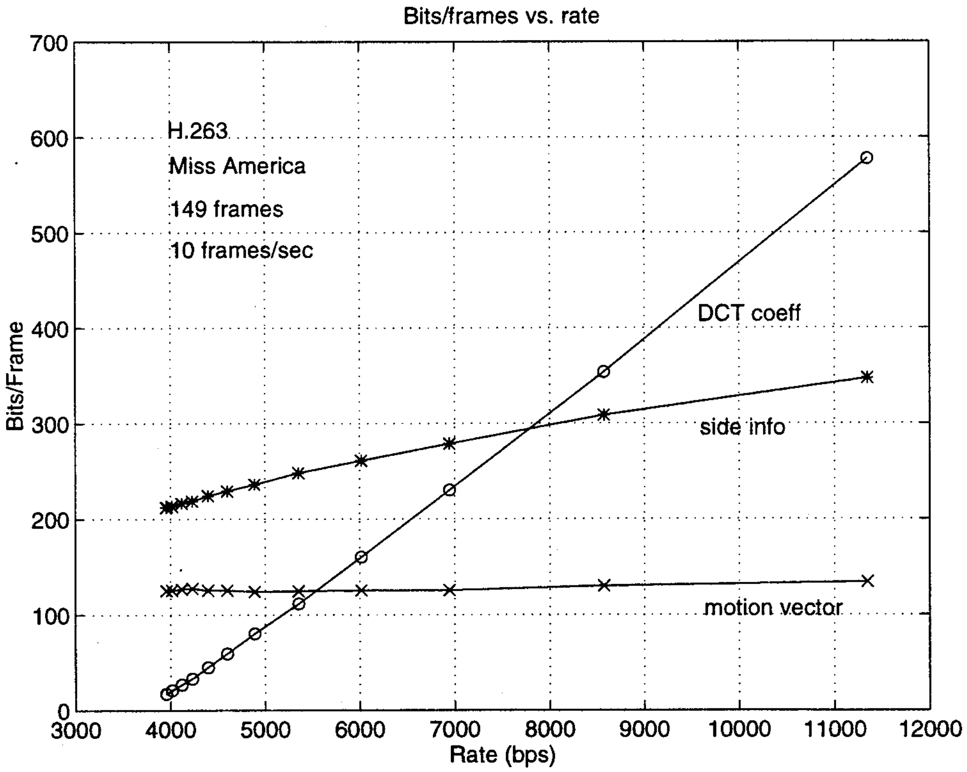


Figure 1: Motion vector bit rate, DCT bit rate, and side information for Telenor's coder.

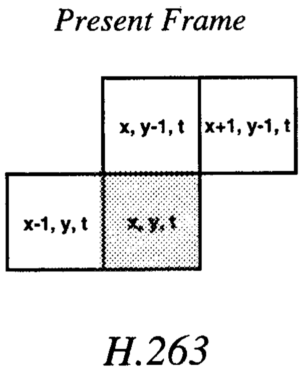


Figure 2: The H.263 region of support used to compute the median.

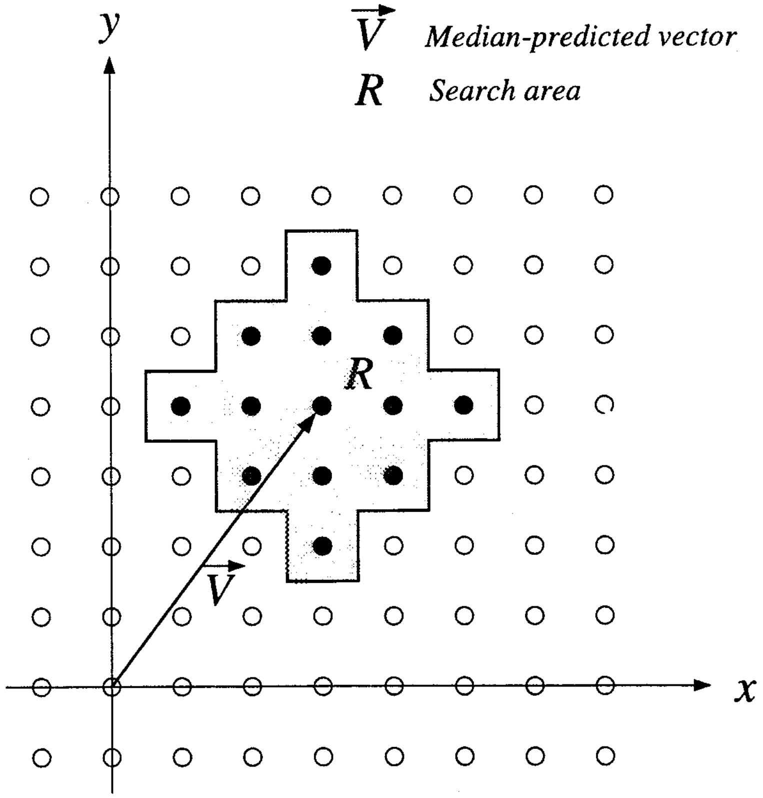


Figure 3: The search area used during motion estimation.

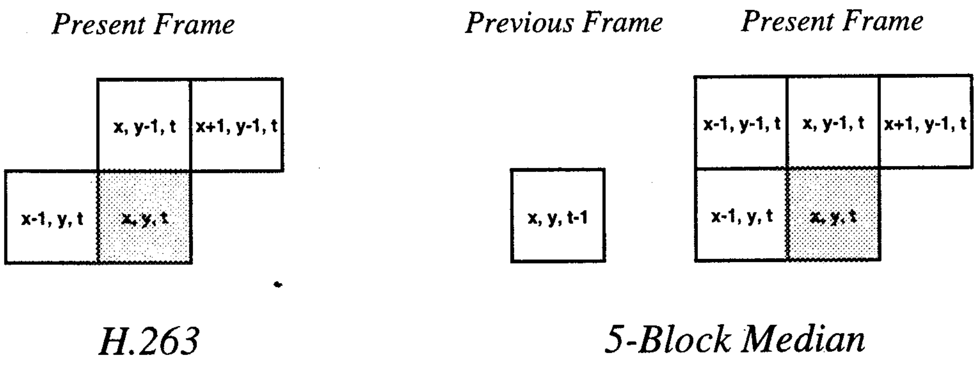


Figure 4: The regions of support used by H.263 and our coder.

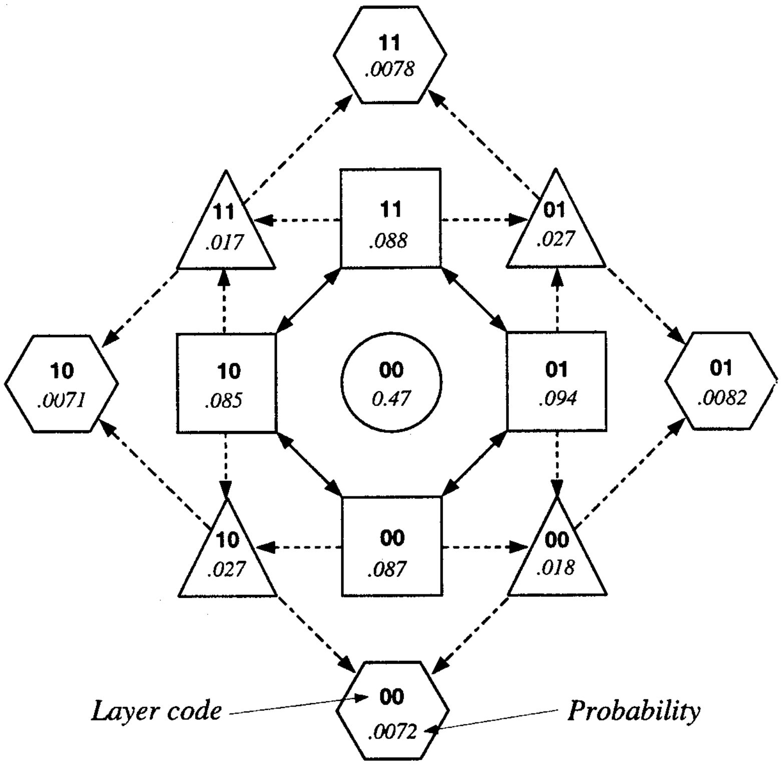


Figure 5: A semi-fixed-length coding method

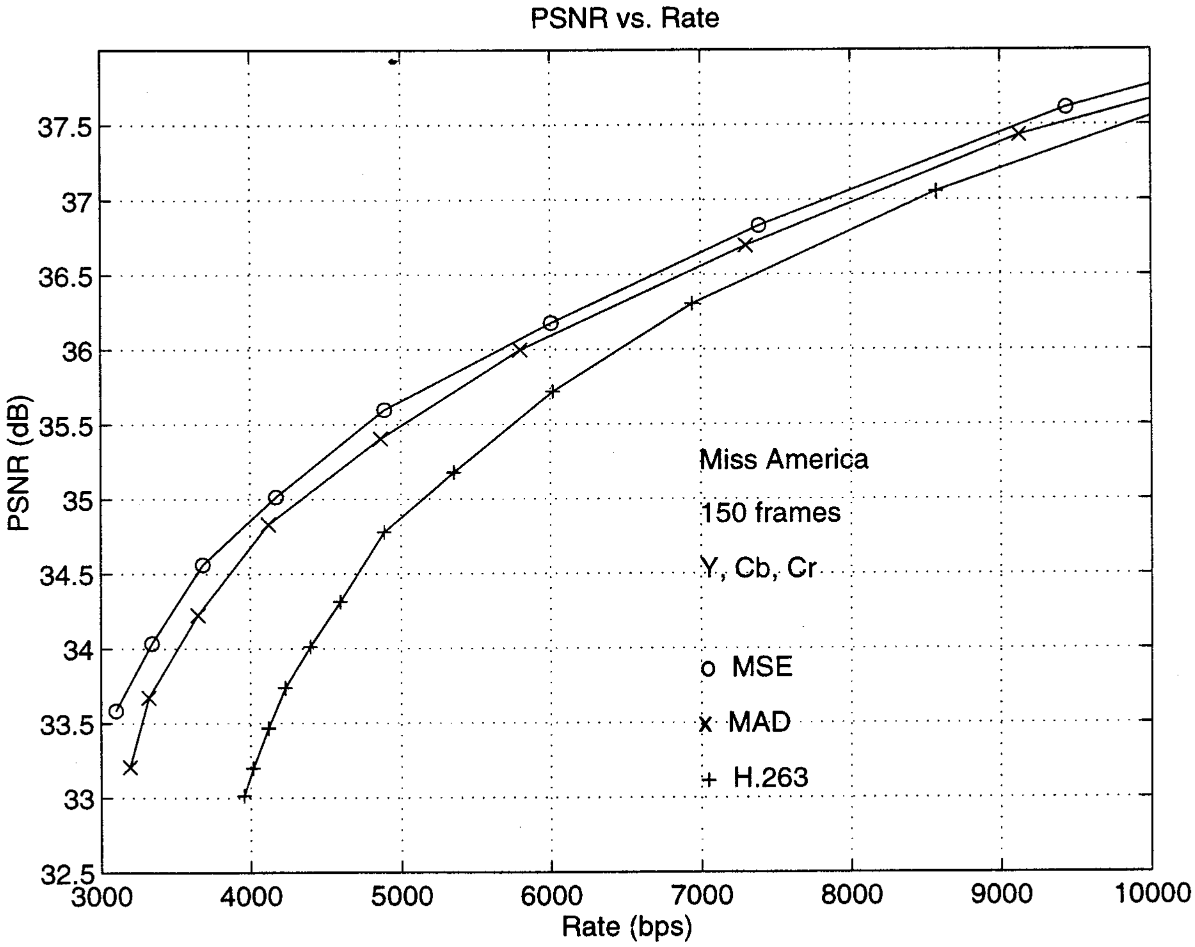


Figure 6: The PSNR improvement due to the rate-distortion-based MB coding control strategy. Both the MSE and the MAD distortion measures are used.

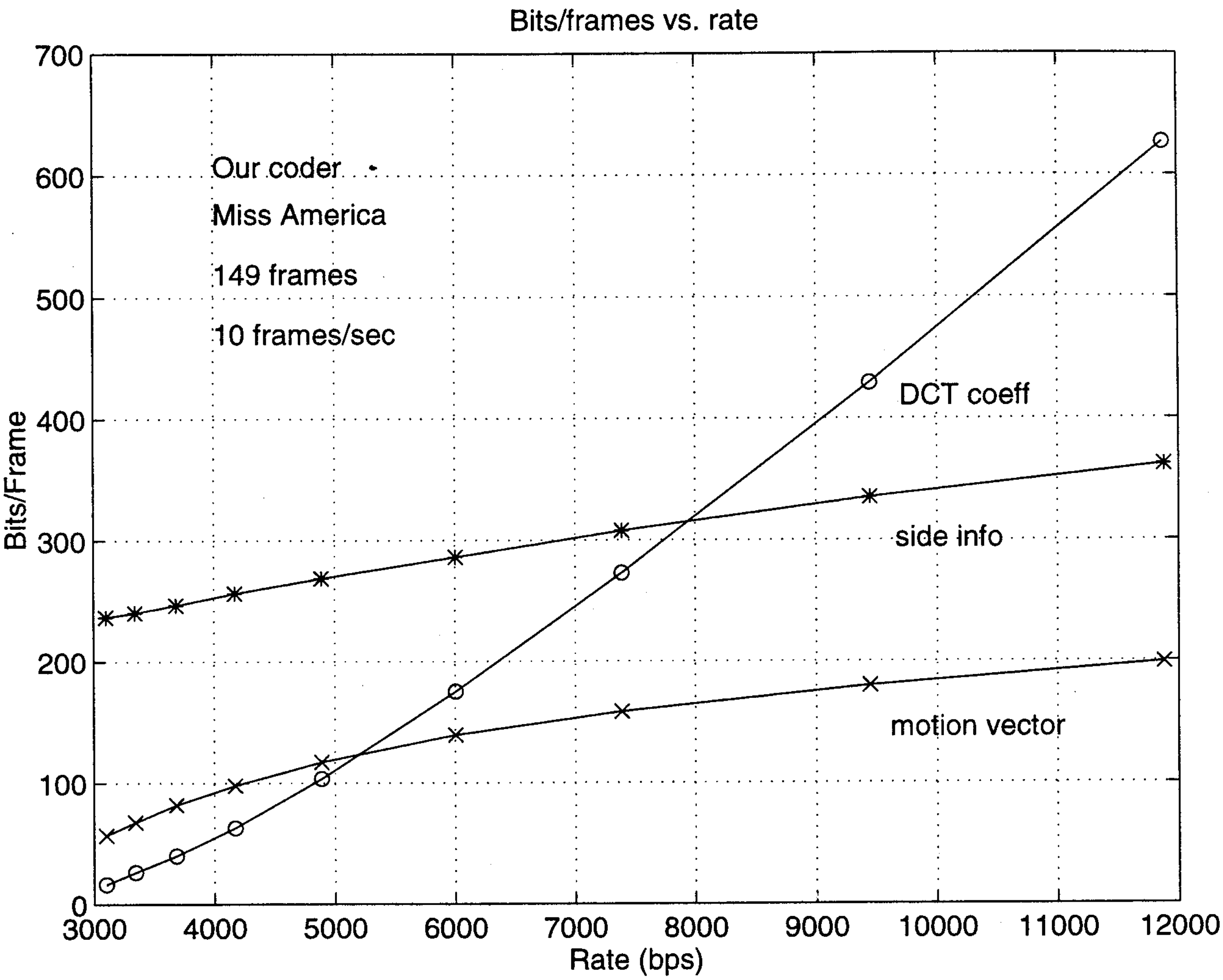


Figure 7: Motion vector bit rate, DCT bit rate, and side information for our coder.

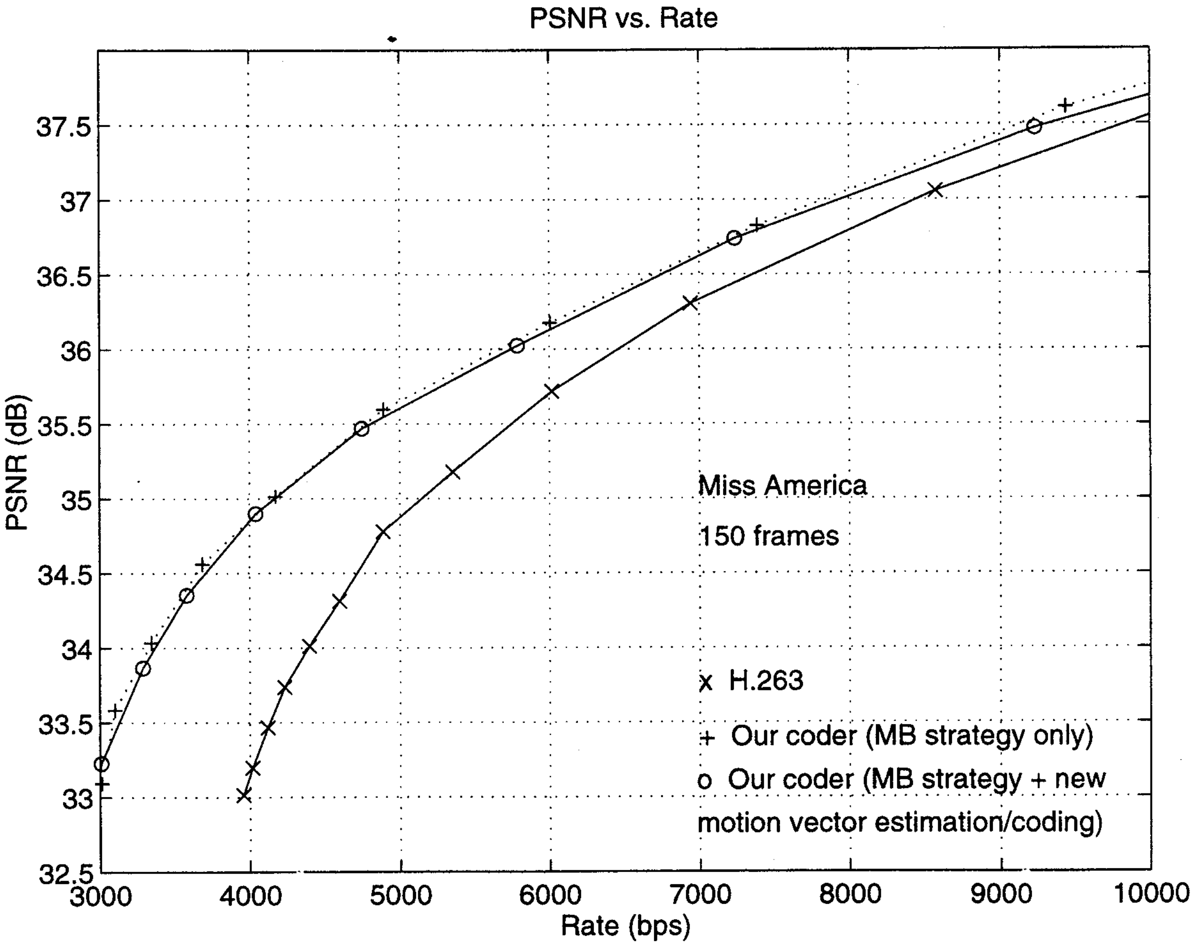


Figure 8: The PSNR loss due to the new motion vector estimation/coding method.

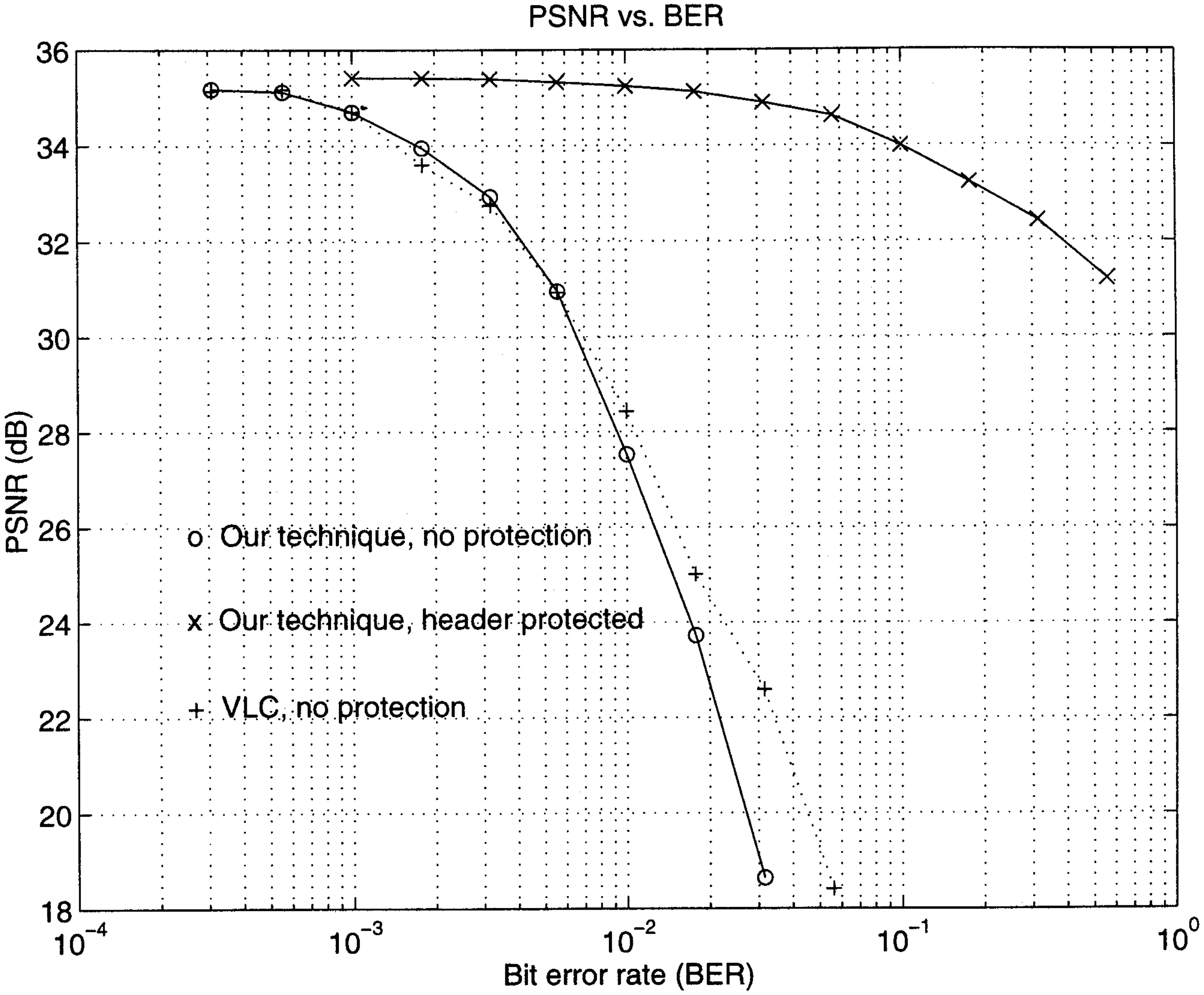


Figure 9: The PSNR as a function of the bit error rate for our semi-fixed-length coding technique and the H.263 VLC technique.

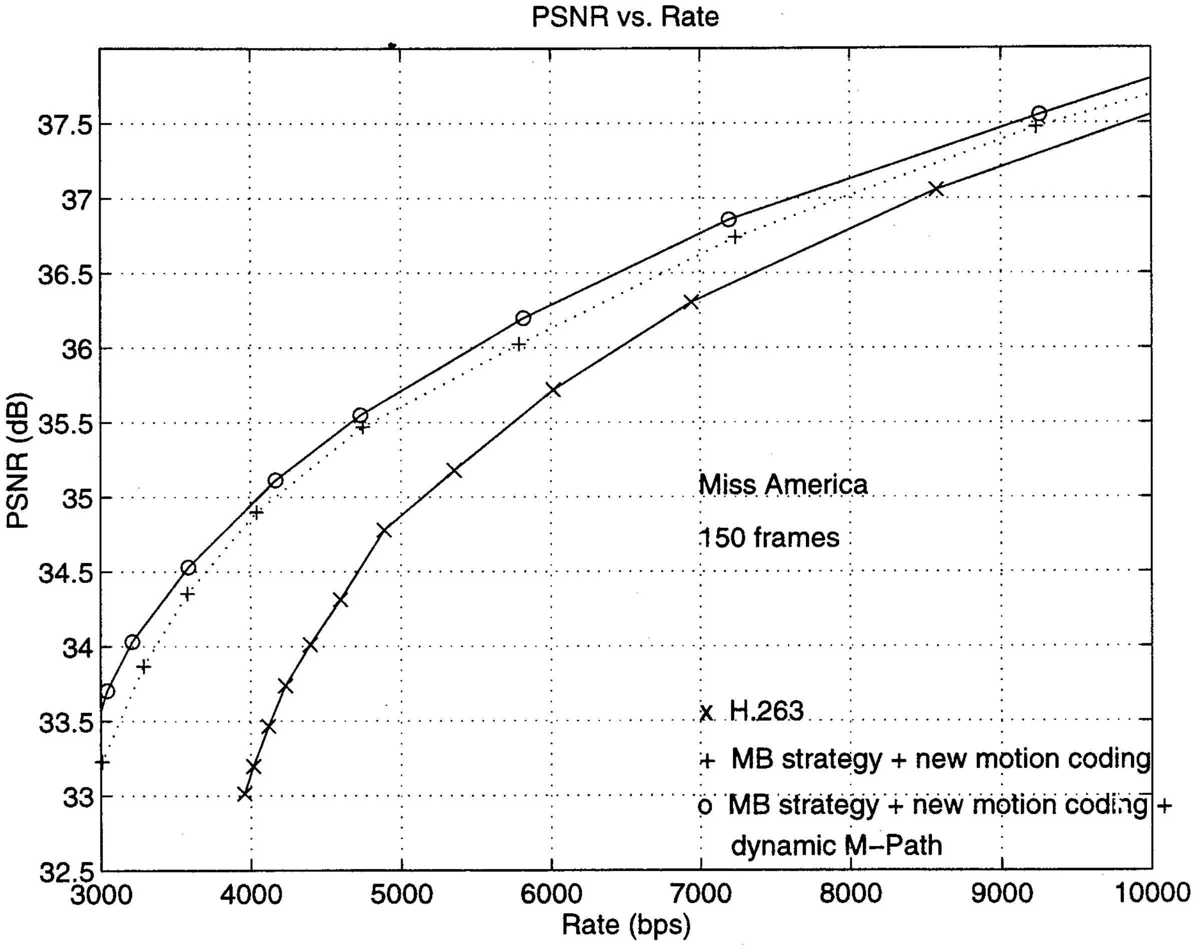
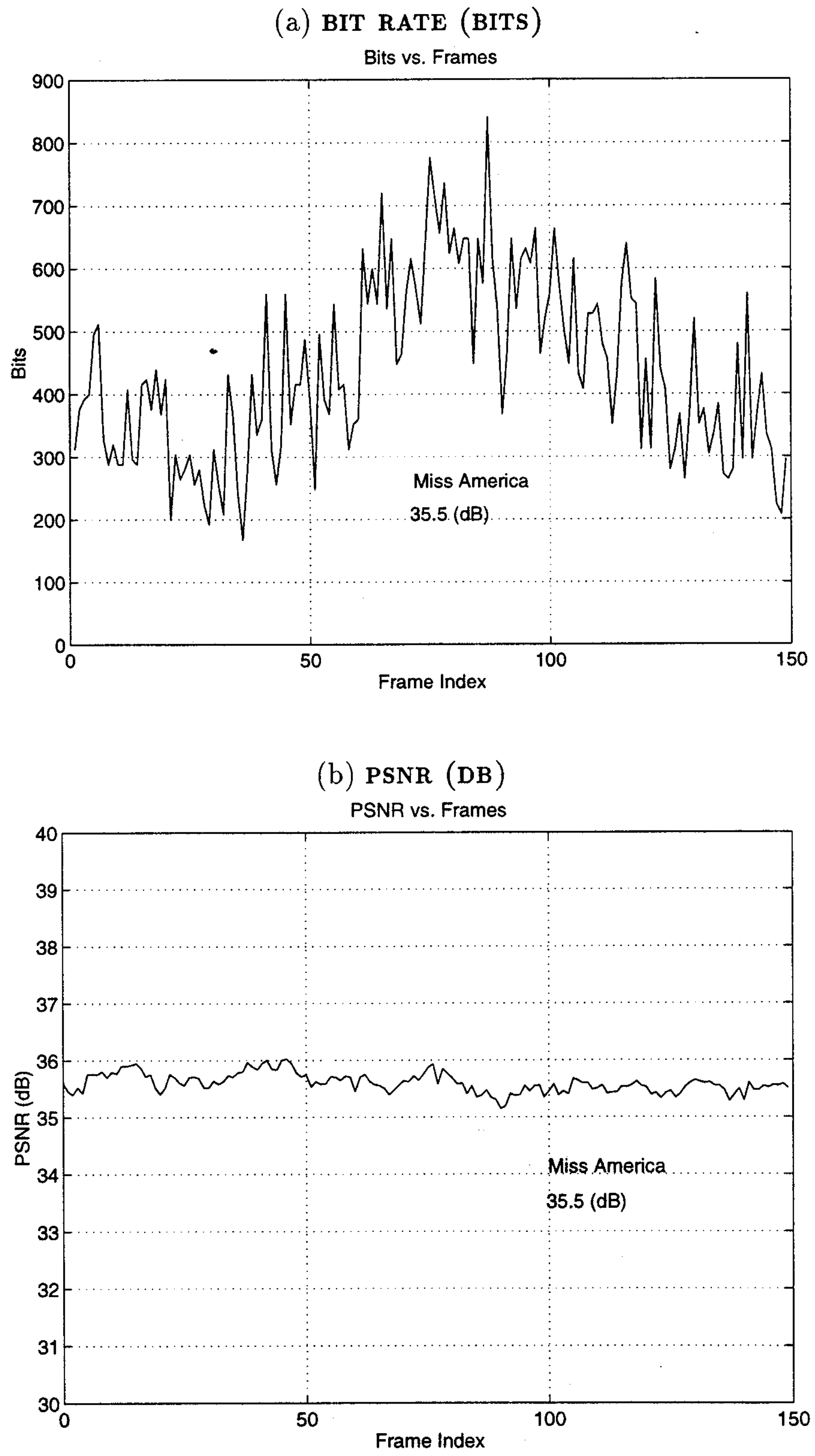
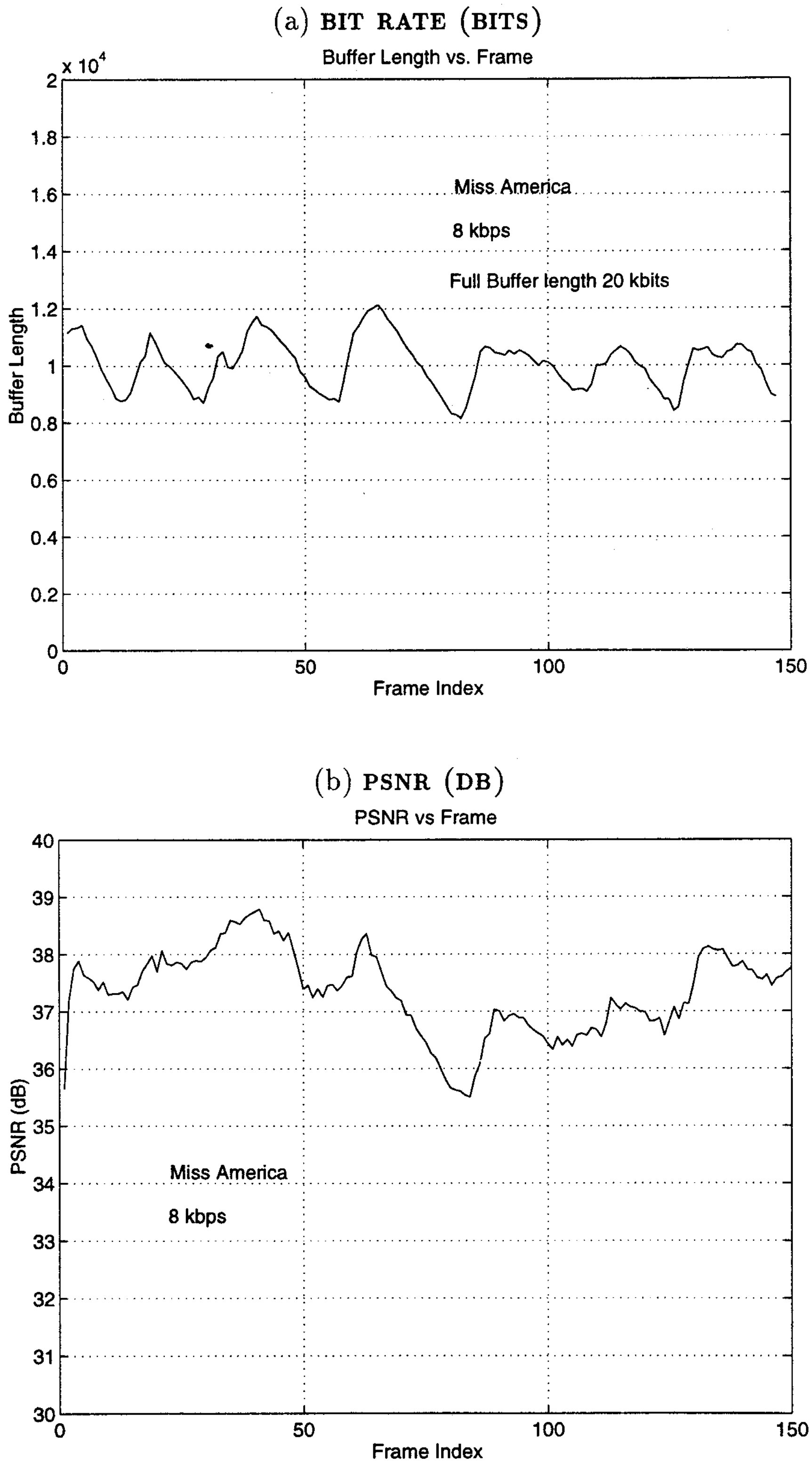


Figure 10: The PSNR improvement due to the dynamic M-search technique.


(a) BIT RATE (BITS)

(b) PSNR (DB)

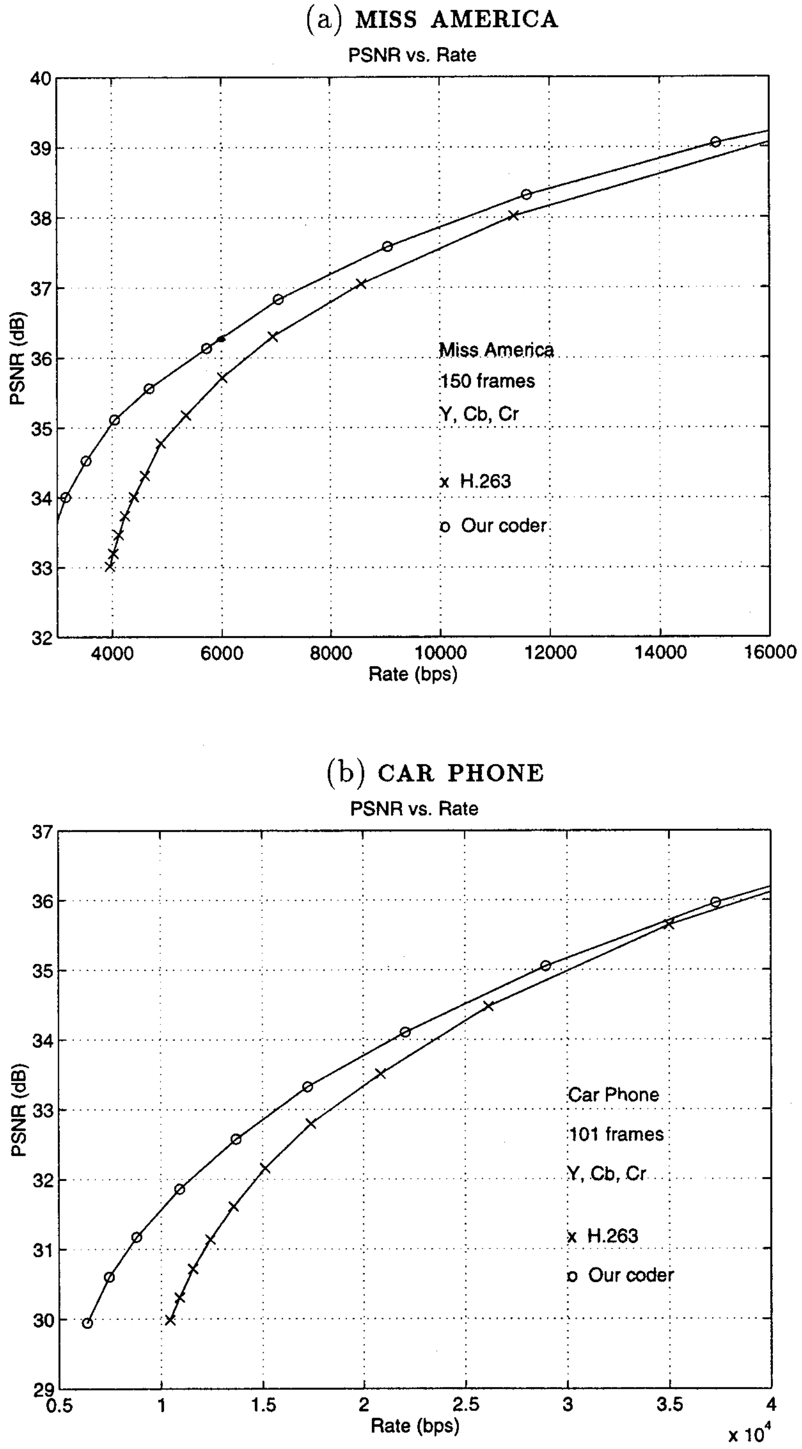
Figure 11: The rate-distortion profile for our coder: (a) Bit rate and (b) PSNR for 150 frames of the test sequence **MISS AMERICA**. The distortion is controlled through varying the parameter λ.



(a) BIT RATE (BITS)

(b) PSNR (DB)

Figure 12: The rate-distortion profile for our coder: (a) Bit rate and (b) PSNR for 150 frames of the test sequence **MISS AMERICA**. The buffer state is used to control the bit rate through varying the parameter λ.



(a) **MISS AMERICA**

(b) **CAR PHONE**

Figure 13: Overall comparison between our coder and Telenor's coder for the sequences (a) **MISS AMERICA** and (b) **CAR PHONE** at bit rates between 3 and 16 kbps, and between 5 and 40 kbps, respectively. the performance gain.